Parameterized Complexity News

Newsletter of the Parameterized Complexity Community

www.fpt.wikidot.com Vol 14, No 1. February 2018
ISSN 2203–109X

Editors Frances Rosamond (Univ Bergen) Frances.Rosamond@uib.no and Valia Mitsou (Univ Paris Diderot) vmitsou@liris.cnrs.fr.

Congratulations to all for many awards and prizes, graduates, new jobs, and wonderful research. This newsletter features an article by PACE prize-winner Hisao Tamaki (Meiji University, JP) on *Positive-instance driven dynamic programming for treewidth*, and an article on *Erdős-Pósa property of chordless cycles and its applications by* Eun Jung Kim (Université Paris-Dauphine) and O-joung Kwon (Incheon National University). Follow fb page @MikeFellowsFPT.

Nerode Prize DEADLINE March 1.

Send a brief summary of the technical content of the nominated paper and a brief explanation of its significance to Dániel Marx (dmarx@cs.bme.hu) Award Committee Chair (Hungarian Academy of Sciences). The Subject line of the nomination E-mail should contain the group of words "Nerode Prize Nomination".

Abu-Khzam awarded CEDRE grant

Congratulations to **Faisal N. Abu-Khzam** (Lebanese American Univ) for his project *Efficient Algorithms for repairing quality in dynamic networks*. The grant should cover 10 research visits, over two years, between LAU and University of Paris-Dauphine. His partners at Dauphine are **Joyce El-Haddad** and **Cristina Bazgan**. The project is funded by CEDRE, the Hubert Curien Partnership (PHC) Franco-Lebanese. It is implemented in France by the Ministry of Europe and Foreign Affairs (MEAE) and the Ministry of Higher Education, Research and Innovation (MESRI). In Lebanon, by the Ministry of Education and Higher Education.

Michael Fellows awarded Research Council of Norway Toppforsk Grant

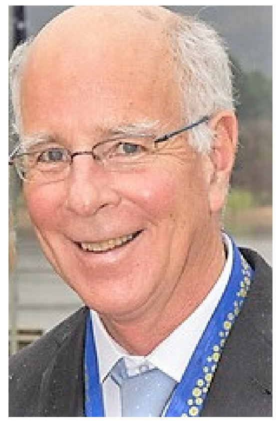

Figure 1: Michael Fellows, University of Bergen

Congratulations to **Michael Fellows.** The Research Council of Norway has awarded a Toppforsk grant of about NOK 25 million for Fellows' project, *Parameterized Complexity for Practical Computing*. The funding scheme supports "scientific quality at the forefront of international research; boldness in scientific thinking and innovation". Mike is an Elite Professor at the University of Bergen (one of five). He was chosen and is supported by the Bergen Research Foundation as well as the university and the Research Council of Norway.

Contents of this issue:

Rod Downey gives Gödel Lecture/wins Shoenfield Prize

Congratulations to **Rod Downey** (Victoria University of Wellington, NZ) who will give the *Gödel Lecture* at The Logic Colloquium 2018, the annual European summer meeting of the *Association of Symbolic Logic (ASL)*, that will be held during July 23–28, 2018 at the University of Udine, Italy.

Congratulations to Rod also for winning the **2016 Shoenfield Prize** for his book with Denis Hirschfeldt: *Algorithmic Randomness and Complexity* (Theory and Applications of Computability, Springer-Verlag New York, 2010).

Matthias Mnich awarded DFG grant

Congratulations to **Matthias Mnich** (Maastricht Univ, Universität Bonn) who has received a personal grant of 578.000 Euro from DFG - Deutsche Forschungsgemeinschaft for his project *Multivariate Algorithms for Scheduling Problems*. Dr. Mnich is also directing *Kernelization for Big Data*, a DFG project associated with the DFG Big Data.

Erdős-Pósa property of chordless cycles and its applications

by Eun Jung Kim (Université Paris-Dauphine) and O-joung Kwon (Incheon National University). eunjungkim78@gmail.com, ojoungkwon@gmail.com

This short article summarizes the idea of the SODA 2018 paper by Eun Jung Kim and O-joung Kwon [6].

Introduction The Erdős-Pósa property of a set of graph patterns $\mathcal{C}$ describes a property that there is a gap function f depending only on $\mathcal{C}$ such that a graph contains either $k+1$ disjoint copies of given graph patterns, or a vertex set of size at most $f(k)$ hitting all such graph patterns. As a simple example, in any graph there are either $k+1$ pairwise vertex-disjoint edges (a matching of size $k+1$), or at most $2k$ vertices meeting all edges. Thus, we can say that $\mathcal{C} = \{K_2\}$ has the Erdős-Pósa property. This property trivially holds if the given set of graph patterns is finite; otherwise, it may not hold in general.

A celebrated result by Erdős and Pósa states that the set of cycles has this property with a gap function $f(k) = \mathcal{O}(k \log k)$ [4]. Furthermore, in polynomial time, one can find either $k+1$ vertex-disjoint cycles or a vertex set of size $\mathcal{O}(k \log k)$ that hits all cycles. Using this, we can obtain an approximation algorithm for Feedback Vertex Set of factor $\mathcal{O}(\log \mathsf{opt})$ as follows. From $t = 1$ to $|V(G)|$, we apply the above algorithm, and find a maximum $t = k$ such that the algorithm outputs $k+1$ vertex-disjoint cycles. This means that the optimal solution opt has at least $k+1$ vertices, while the algorithm outputs a set of vertices of size $\mathcal{O}(k \log k) \le \mathcal{O}(\mathsf{opt} \log \mathsf{opt})$ hitting all cycles, by applying the next value of t. Since there is a 2-approximation algorithm for Feedback Vertex Set, the above approximation algorithm is less useful, but for other problems, this simple approach may give a good approximation algorithm.

A *chordless cycle*, also known as a *hole*, in a graph G is an induced subgraph of G which is a cycle of length at least four. A graph is *chordal* if it has no chordless cycles. As a natural variant of the Feedback Vertex Set problem, Marx [7] first studied the Chordal Vertex Deletion problem, which asks given a graph G and an integer k whether there is a set of at most k vertices whose deletion turns G into a chordal graph. He showed that Chordal Vertex Deletion is fixed parameter tractable, and later Cao and Marx [3] obtained a better algorithm that runs in time $2^{\mathcal{O}(k \log k)} poly(n)$. Also, Marx [7] asked whether Chordal Vertex Deletion admits a polynomial kernel or not. This problem has been recently resolved by Jansen and Pilipczuk [5]. One of the main tools in this kernelization algorithm is an approximation algorithm of factor $\mathcal{O}(\mathsf{opt}^2 \log \mathsf{opt} \log n)$. Agrawal et. al. [1] obtained an approximation algorithm with improved factor $\mathcal{O}(\mathsf{opt} \log^2 n)$. The factor of an approximation is intrinsically linked to the quality of the obtained kernels. A factor-$\mathcal{O}(\log^2 n)$ approximation algorithm was proposed more recently by Agrawal *et. al.* [2].

In this context, we studied the question whether the set of holes has the Erdős-Pósa property, that is, whether that there is a function g such that one can find either $k+1$ vertex-disjoint holes in a graph or a vertex set of at most $g(k)$ vertices that hits all holes. This was asked by Jansen and Pilipczuk [5].

We prove that the set of holes has the Erdős-Pósa property. Explicitly,

($*$) one can in polynomial time find either $k+1$ vertex-disjoint holes, or $ck^2 \log k$ vertices hitting every hole for some constant c.

As expected, this result gives an approximation algorithm for Chordal Vertex Deletion of factor $\mathcal{O}(\mathsf{opt} \log \mathsf{opt})$.

Proof sketch We reduce the statement ($*$) to the following statement:

($**$) for a given graph G, a shortest hole C in G and an integer k such that $G - V(C)$ is chordal, one can in polynomial time find either $k+1$ vertex-disjoint holes, or $\mathcal{O}(k \log k)$ vertices hitting every hole.

Suppose G and k are given for the statement ($*$). Let $G_1 := G$, and we recursively find a shortest hole C_i in G_i and set $G_{i+1} = G_i - V(C_i)$ unless G_i has no holes. Let $G_1, G_2, \ldots, G_m, G_{m+1}$ be the obtained sequence. If $m \ge k+1$, then $C_1, C_2, \ldots, C_{k+1}$ are $k+1$ vertex-disjoint holes. Therefore, we may assume $m \le k$. Now, we apply ($**$) for the pair (G_m, C_m). The algorithm in ($**$) outputs either $k+1$ vertex-disjoint holes or a vertex set S_m of size $\mathcal{O}(k \log k)$ hitting all holes in G_m. In the latter case, we again apply ($**$) for the pair $(G_{m-1} - S_m, C_{m-1})$, and obtain either $k+1$ vertex-disjoint holes or a vertex set S_{m-1} of size $\mathcal{O}(k \log k)$ hitting all holes in $G_{m-1} - S_m$. If we proceed from G_m to G_1, then either we obtain $k+1$ vertex-disjoint holes or the vertex set $S_1 \cup S_2 \cup \cdots \cup S_m$ of size $\mathcal{O}(k^2 \log k)$ hits all holes in G. Therefore, we may focus on proving $(**)$.

Suppose (G, C, k) is a given tuple for $(**)$. One can observe several structural properties related to the shortest hole C. We may first assume that C has length at least $c'k \log k$ for some constant c'; otherwise, we can just take all vertices in C because $G - V(C)$ is chordal. Also, we can have the following: (1) For a vertex v in $V(G) \setminus V(C)$ having a neighbor in C, either v dominates C or it has at most 3 neighbors that are consecutive. (2) Set D as the set of all vertices in $V(G) \setminus V(C)$ dominating C. Then D is a clique. (3) For $v \in V(C)$, set $Z_v := (N(v) \setminus V(C) \setminus D) \cup \{v\}$ where $N(v)$ denotes the set of neighbors of v. If the distance between two vertices $v, w \in V(C)$ in C is at least 4, then there are no edges between Z_v and Z_w. (4) For a subgraph H contained in the closed neighborhood of C, set $\text{supp}(H) := \{v \in V(C) : Z_v \cap V(H) \neq \emptyset\}$, and $C[\text{supp}(H)]$ is called the support of H. If H is connected, then its support is also connected. This is because of (3).

We classify holes into four types. A hole in G is a *sunflower* if it is contained in the closed neighborhood of C and a *tulip* otherwise. A hole in G is D-*traversing* if it contains a vertex of D, and D-*avoiding* otherwise. Combining these types, we get four types of holes. For each type of holes, we show that one can in polynomial time find either $k+1$ vertex-disjoint holes or a vertex set of size $\mathcal{O}(k \log k)$ hitting all holes of the special type.

We briefly explain why we classify the types of holes as above. Imagine that there is a hole Q other than C, and we want to hit Q. As Q is a cycle, the intersection points between Q and C partition Q into several subpaths unless Q and C meet at exactly one vertex. If one of these subpaths P contains a vertex outside the closed neighborhood of C, then P with a subpath of C between the end points of P contains a hole. With a more involved argument, we can find $k+1$ vertex-disjoint holes or a vertex set of size $\mathcal{O}(k \log k)$ hitting the end vertices of such paths, and thus in the latter case, we hit the original hole Q. This is a basic strategy to obtain the Erdős-Pósa result for D-avoiding tulips. We use an argument similar to Gallai's A-path Theorem.

When all subpaths are contained in the closed neighborhood of C, or equivalently Q is contained in the closed neighborhood of C, we need totally different approach. We focus on D-avoiding case. We first greedily detect holes having support on at most 7 vertices. We can show that either there are $k+1$ disjoint holes, or a vertex subset S of C of at most $19k$ vertices hitting all sunflowers with support on at most 7 vertices. Somewhat surprisingly, all remaining sunflowers in $G - S$ must have support exactly $V(C)$. For contradiction, suppose that there is a sunflower Q in $G - S$ having support on more than 7 vertices but not $V(C)$. By (4), the support of Q is connected, and therefore, it is a path on C, say from x to y. We choose $a \in Z_x \cap V(Q)$ and $b \in Z_y \cap V(Q)$ and let P_1 and P_2 be the two subpaths of Q from a to b. As the supports of P_1 and P_2 are also connected, if Q contains a vertex on C having distance at least 3 to $\{a, b\}$ in C, then it has a neighbor in an internal vertex of one of P_1, P_2, which is impossible. Thus without loss of generality we may assume $V(Q) \cap V(C)$ contains a vertex which have distance at most 2 to a in C. Let c be the vertex in the support having distance 4 to a in C. Then $c \notin V(Q)$ and c has neighbors in both P_1 and P_2. Thus together with c and subpaths of P_1 and P_2 from q to the neighbors of c in P_1 and P_2, respectively, one can find a hole having support on at most 7 vertices, containing a vertex in $V(Q) \cap V(C)$. But this is contradiction, as we hit all such holes by S.

The remaining task is to obtain the Erdős-Pósa property for sunflowers with support exactly $V(C)$. For those holes, we choose 4 vertices a, b, c, d of C (in this order) whose pairwise distance are at least some constant, and there are no vertices of S in the subpaths from a to b and from c to d. By Menger's Theorem, we find a set $\mathcal{P}$ of $k+13$ vertex-disjoint paths from Z_a to Z_d following the one direction of C, and a set $\mathcal{Q}$ of $3k+15$ vertex-disjoint paths from Z_b to Z_c following the different direction of C. If one of them does not exist, then Menger's Theorem gives a small vertex set hitting all such paths, thus hitting all remaining sunflowers. Otherwise, we can pair up paths of $\mathcal{P}$ and $\mathcal{Q}$ and construct $k+1$ disjoint holes. Briefly speaking, the non-existence of a hole with small support provides that for any pair $P \in \mathcal{P}$ and $Q \in \mathcal{Q}$, there is an edge between P and Q near Z_a and also Z_d.

D-intersecting holes Q have special properties; they contain exactly one vertex of D, and at most two vertices of C. Using this basic observation and the König's Theorem on bipartite graphs, we obtain the Erdős-Pósa property for D-intersecting holes.

Future work It would be interesting to see whether the bound $\mathcal{O}(k^2 \log k)$ on gap function can be improved to $\mathcal{O}(k \log k)$. A lower bound $\Omega(k \log k)$ follows from the lower bound for usual cycles [4]. We also show that holes of length at least ℓ for any $\ell \geq 5$ have no Erdős-Pósa property. We ask whether holes of length at least ℓ for any $\ell \geq 6$ have the Erdős-Pósa property in C_4-free graphs.

References

[1] A. Agrawal, D. Lokshtanov, P. Misra, S. Saurabh, and M. Zehavi. Feedback vertex set inspired kernel for chordal vertex deletion. In *Proc. SODA2017*, pages 1383–1398, 2017.

[2] A. Agrawal, D. Lokshtanov, P. Misra, S. Saurabh, and M. Zehavi. Polylogarithmic approximation algorithms for weighted-$\mathcal{F}$-deletion problems. *CoRR*, abs/1707.04908, 2017.

[3] Y. Cao and D. Marx. Chordal editing is fixed-parameter tractable. *Algorithmica*, 75(1):118–137, 2016.

[4] P. Erdős and L. Pósa. On the independent circuits contained in a graph. *Canad. J. Math.*, 17:347–352, 1965.

[5] B. M. P. Jansen and M. Pilipczuk. Approximation and kernelization for chordal vertex deletion. In *Proc. SODA2017*, pages 1399–1418, 2017.

[6] E. J. Kim and O. Kwon. Erdős-pósa property of chordless cycles and its applications. In *Proc. SODA2018*, pages 1665–1684, 2018.

[7] D. Marx. Chordal deletion is fixed-parameter tractable. *Algorithmica*, 57(4):747–768, 2010.

Positive-instance driven dynamic programming for treewidth

by Hisao Tamaki (Meiji University, JP). tamaki@cs.meiji.ac.jp.

This short article summarizes the work reported in the ESA 2017 Track B Best Paper Award paper by myself [6]. I focus on the overall scenario in which a theoretical algorithm is turned into a practically efficient algorithm. For technical details, please see the ESA paper or its arxiv version. I assume the readers' familiarity with the basic graph concepts and notation as well as the definition of treewidth.

Dynamic programming algorithm of Bouchitté and Todinca

Let G be a graph. A vertex set of S is a *separator* of G, if there are two vertices a and b that belong to the same connected component of G but to distinct components of $G[V(G) \setminus S]$. A *block* of G is a pair (S, C), where S is a separator and C is a connected component of $G[V(G) \setminus S]$. Block (S, C) is *full* if S is the neighborhood of C in G. A separator S of G is a *minimal separator* if there are at least two distinct full blocks of the form (S, C). We say that block (S, C) is *minimally separated* if S is a minimal separator. Note that a minimally separated block is necessarily full.

We consider the decision version of the treewidth problem: given a graph G and a positive integer k, decide if $\mathrm{tw}(G)$, the treewidth of G, is at most k. In this problem instance, a separator S (or a block (S, C)) is *relevant* if $|S| \leq k$. We say that block (S, C) is *feasible* in this instance, if there is a tree decomposition of $G[S \cup C]$ of width k or smaller that has a bag containing S. This last condition ensures that this tree decomposition of $G[S \cup C]$ can be used as a subtree of the tree decomposition of entire G.

The classical dynamic programming algorithm due to Arnborg, Corneil, and Proskurowski [1] for treewidth lists all relevant full blocks and then inductively determines if each block is feasible. In the decision problem version of the dynamic programming algorithm due to Bouchitté and Todinca [3], only the feasibility of relevant minimally separated blocks are considered. This is a big theoretical leap in treewidth computation which has lead to polynomial time algorithms for many graph classes and to the best known running time upper bound for general graphs [5]. The gap between the number of relevant full blocks and that of relevant minimally separated blocks is indeed huge, unless the graph is tiny or k is very small. Their algorithm, however, had not been considered competitive in practice prior to my work. To see why, let us look at the actual numbers of combinatorial objects involved in the execution of their algorithm for some instances.

The recurrence defining feasible minimally separated blocks involves *potential maximal cliques*: a vertex set S of G is a potential maximal clique of G if there is some minimal triangulation of G in which S is a clique. In the dynamic programming for input G and k, we need potential maximal cliques of G of cardinality $k + 1$.

Table 1 lists those numbers for graphs generated from the $G(n, m)$ model: given n and m, a graph is drawn uniformly at random from the set of all graphs with n vertices and m edges. A single instance is examined for each pair (n, m). Both the total number and the number of relevant minimal separators, where k is set to the treewidth of the graph, are listed and the same applies to potential maximal cliques.

			min. separators		pot. max. cliques	
$\lvert V\rvert$	$\lvert E\rvert$	tw	all	$\leq$ tw	all	$\leq$ tw +1
20	40	6	98	51	376	115
20	60	8	191	48	796	96
20	80	11	185	122	698	376
20	100	11	107	25	354	37
30	60	7	535	185	3122	559
30	90	11	2983	247	20154	682
30	120	14	2713	376	16736	1137
30	150	16	1913	281	10535	768
40	80	8	14842	1070	178661	5341
40	120	14	164773	2356	1740644	10372
40	160	18	134485	3952	1251656	17360
40	200	20	52182	1790	423691	6820
50	100	10	96499	1361	1123621	6029
50	150	16	1792713	9152	>2000000	48068
50	200	20	2130811	7878	>2000000	36388
50	250	24	1452449	10571	>2000000	47729

Table 1: The numbers of combinatorial objects in treewidth computation

From this table, we can see that the number of relevant objects grows much slower than the number of all objects. This suggests the advantage of using decision problems for optimization: the naive approach of trying $k = 1, \ldots$ until the answer is YES is probably the best, since the sum of the numbers of relevant objects for all these values of k is still much smaller than the number of all objects.

There is a caveat, however. That the number of objects is small does not necessarily mean that they can be generated quickly. Indeed, we do not know efficient methods for generating relevant minimal separators and relevant potential maximal cliques. Although it is known that all the minimal separators of a given graph with n vertices can be generated in time polynomial in n per each [2] and that all the potential maximal cliques can be generated in time polynomial in the number of minimal separators [4], similar results are not known for generating those vertex sets with a cardinality constraint. This is the main obstacle in implementing the Bouchitté-Todinca dynamic programming with practical efficiency.

Positive-instance driven approach

My approach in [6] was to design a positive-instance driven (PID) variant of the the Bouchitté-Todinca dynamic programming: rather than listing all relevant minimally separated blocks (together with all relevant potential maximal cliques) and then deciding the feasibility of each block in the list in a bottom up manner, it generates only feasible minimally separated blocks, building each on smaller feasible ones. Our implementation (with Hiromu Ohtsuka) of the algorithm turned out quite competitive. It solved a number of standard benchmark instances for which the optimal solutions had not previously been known and, combined with some preprocessing techniques, solved all the 200 instances posed by PACE 2017 [7] for the exact treewidth track within the allocated time of 30 minutes each.

Initially, I thought the success was due to the gap between the number of all relevant minimally separated blocks and that of feasible ones – the gap the PID approach was meant to exploit. I have learned through the experiments, however, that this gap is rather small in many instances. My current conclusion is that the success of the approach mainly lies in overcoming, to some extent, the difficulty mentioned in the last section: it provides a practically efficient way of generating relevant minimal separators and potential maximal cliques. I emphasize the possibility that a standard (non-PID) version of the Bouchitté-Todinca dynamic programming outperforms the PID variant if a more efficient algorithm for generating relevant minimal separators and potential maximal cliques is developed in the future.

References

[1] S. Arnborg, D. G. Corneil, and A. Proskurowski: Complexity of finding embeddings in a k-tree. SIAM Journal on Algebraic Discrete Methods 8, 277-284, 1987

[2] A. Berry, J.-P. Bordat, and O. Cogis: Generating all the minimal separators of a graph. International Journal of Foundations of Computer Science 11(03), 397-403, 2000.

[3] V. Bouchitté and I. Todinca: Treewidth and minimum fill-in: Grouping the minimal separators. SIAM Journal on Computing 31(1), 212-232, 2001

[4] V. Bouchitté and I. Todinca: Listing all potential maximal cliques of a graph. Theoretical Computer Science 276, 17-32, 2002

[5] F. Fomin and Y. Villanger: Treewidth computation and extremal combinatorics. Combinatorica 32(3), 289-308, 2012

[6] H. Tamaki. Positive-instance driven dynamic programming for treewidth. *ESA 2017*, 68:1-68:13, 2017.

[7] PACE 2017 website: https://pacechallenge.wordpress.com/

IPEC 2018, 22-24 August in Helsinki

The 13th International Symposium on Parameterized and Exact Computation (IPEC 2018) covers research in all aspects of parameterized and exact algorithms and complexity.

IPEC 2018 will be part of ALGO 2018, which also hosts ESA 2018 and a number of more specialized conferences and workshops. IPEC 2018 will take place 22-24 August 2018, in Helsinki, Finland.

Accepted papers will be published in the symposium proceedings in the Leibniz International Proceedings in Informatics (LIPIcs) series, based at Schloss Dagstuhl. Authors of accepted papers are expected to present their work at the symposium, and to incorporate the comments from the program committee. A journal special issue is planned for selected papers presented at IPEC 2018.

IMPORTANT DATES: Note that the submission deadline is earlier this year. Title and short abstract registration deadline: May 14, 23:59 AoE
Full paper submission deadline: May 17, 23:59 AoE
Notification: July 1

AWARDS: The Program Committee may award one or more Best Paper Award(s) and Best Student Paper Award(s). Advise the Program Committee if you are submitting for a **Best Student Paper Award**. A student is someone who has not received a PhD degree before the full paper submission deadline. A paper accepted to the conference is eligible for the Best Student Paper Award if either all its authors are students, or if there is one non-student co-author that confirms that a clear majority of conceptual work on the paper was done by the student co-author(s). It is expected that a student gives a presentation at the conference.

SPECIAL EVENTS

NERODE PRIZE: An invited talk will be given by the 2018 EATCS-IPEC Nerode Prize winner.

TUTORIAL: Radu Curticapean will give an invited tutorial on *Counting Problems in Parameterized Complexity*.

PACE: There will be a session presenting the results of the *3rd Parameterized Algorithms and Computational Experiments Challenge* (PACE 2018).

PACE Register your team now

JOIN THE CHALLENGE. The challenge consists of three tracks on the Steiner Tree problem. Detailed instructions about the three tracks, their rules, the instance sets, and the input and output formats: Challenge https://pacechallenge.wordpress.com/pace-2018/

Download the public instances on the PACE challenge website and start your implementation. Test your code on this platform now: https://pacechallenge.wordpress.com/2017/12/12/optil-io/ .

TIMELINE:
–May 1st, 2018: Submission of final program
–May 10th, 2018: Result are communicated to participants
–August 20-24 2018: Award ceremony at the International Symposium on Parameterized and Exact Computation (IPEC 2018) in Helsinki

Prizes and travel awards of 4000 Euros have been provided for participants through the generous sponsorship of NETWORKS (http://thenetworkcenter.nl/), an NWO Gravitation project of the Univ of Amsterdam, Eindhoven Univ of Technology, Leiden Univ, and the Center for Mathematics and Computer Science (CWI).

FPT Data Wrangling for Social Good–Wellington, New Zealand–July 25

This FPT workshop immediately follows the 4th Creative Mathematical Sciences Communication (CMSC 2018) conference which will be held 21-23 July in Wellington. Come for both. For details contact organizers Catherine McCartin (c.m.mccartin@massey.ac.nz) and Peter Shaw (peter.shaw.cs@gmail.com), both at Massey Univ.

4th Creative Mathematical Sciences Communication conference (CMSC2018)

The 4th Creative Mathematical Sciences Communication (CMSC 2018) conference will be held 21-23 July in Wellington, NZ (website is http://www.cmsc.nz). Stay for the FPT workshop immediately following.

Join scientists, researchers, teachers, and artists in developing new ways of communicating mathematical and computational thinking. Welcome are contributions in art forms such as dance, graphic art, theatre, and the myriad of ways to communicate science to the public. The conference will feature keynote talks by leading researchers and communicators in the mathematical sciences, sharing their experience, new initiatives, and ideas. The conference will be held in Wellington, New Zealand, at The Learning Connexion (TLC) on 21–23 July 2018. Organizers: Frances Rosamond, Univ Bergen (Frances.Rosamond@uib.no) and Jonathan Milne, The Learning Connexion (jonathanmilnetlc@gmail.com).

Moving Around – Congratulations to all

Dr. Bin Sheng has joined the Nanjing University of Aeronautics and Astronautics. Congratulations, Bin Sheng.
O-Joung Kwon has accepted an assistant professor position at the mathematics department of Incheon National University in South Korea, starting from March 2018. Congratulations O-Joung.
Peter Shaw has joined Massey University, New Zealand. He will be setting up a research group at Massey's Hebut Uni of Technology campus in Tianjin, China. Congratulations, Peter.

CONGRATULATIONS New PhDs

Akanksha Agrawal, Graph Modification Problems: Beyond the Known Boundaries, The University of Bergen, Norway. Advisor: Prof. Saket Saurabh (Univ Bergen, IMSC-Chennai). Co-advisor: Prof. Daniel Lokshtanov (Univ Bergen). Congratulations, Dr. Agrawal. Dr. Agrawal has accepted a post doc position with Dániel Marx.

Figure 2: Dr. Akanksha Agrawal, University of Bergen